Informe del Ejercicio Práctico: Automatización de Pruebas de SQL Injection sobre Varias URLs

1. Objetivo del Ejercicio:

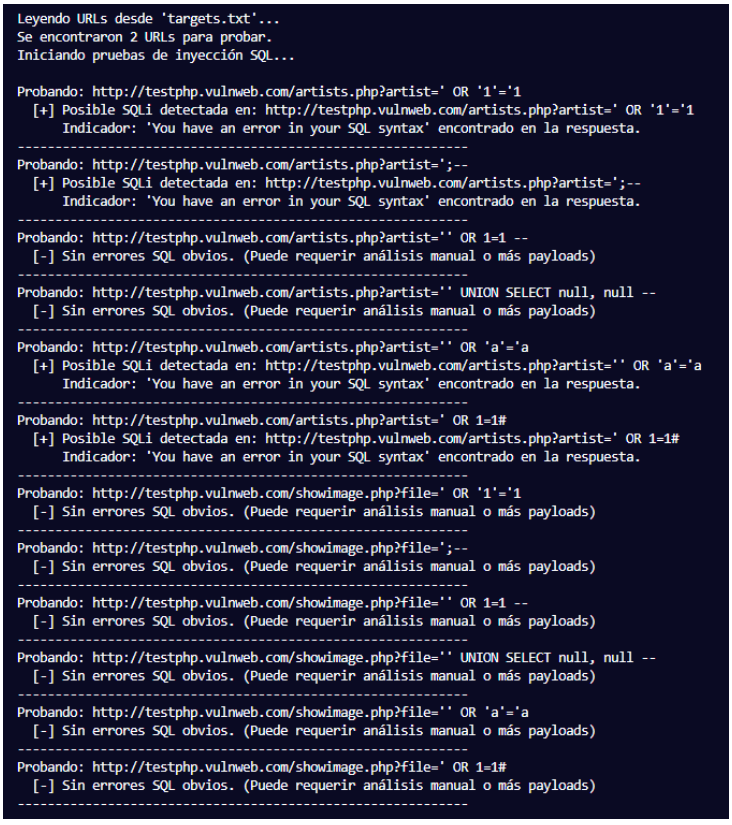
El objetivo principal de este ejercicio fue construir un script en Python para automatizar la detección de posibles vulnerabilidades de SQL Injection en múltiples URLs de un entorno de pruebas. Esto implicó utilizar una lista de payloads para probar distintos parámetros y analizar los comportamientos del servidor que pudieran indicar la presencia de inyección SQL, generando un reporte con los endpoints potencialmente vulnerables.

2. Descripción del Programa:

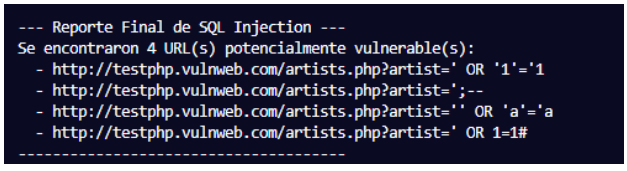
El script detector_sql_injection.py es una herramienta de línea de comandos diseñada para realizar pruebas automatizadas de SQL Injection. Lee un listado de URLs objetivo desde un archivo targets.txt, y para cada URL, itera a través de una lista predefinida de payloads de inyección SQL. Envía solicitudes HTTP GET con los payloads incrustados y analiza las respuestas del servidor en busca de patrones que indiquen errores de base de datos o comportamientos anómalos, comunes en vulnerabilidades de SQL Injection. Los resultados se resumen en un reporte al finalizar las pruebas.

3. Funcionalidades Implementadas:

- **Lectura de URLs desde Archivo:**
    - El script es capaz de leer URLs de forma dinámica desde un archivo targets.txt, lo que permite probar múltiples objetivos sin modificar el código fuente. Se manejan errores si el archivo no existe o está vacío.
- **Gestión de Payloads de SQL Injection:**
    - Utiliza una lista predefinida de payloads clásicos de SQL Injection, enfocados en la detección de errores de sintaxis y la verificación de la lógica booleana (' OR '1'='1, ';--, '' UNION SELECT null, null --, etc.).
- **Inyección y Solicitudes HTTP:**
    - Para cada URL y cada payload, el script construye dinámicamente la URL de prueba inyectando el payload en el parámetro final.
    - Se utiliza la librería requests para realizar las solicitudes HTTP GET, con un timeout para manejar respuestas lentas o sitios no disponibles.



- **Detección de Vulnerabilidades:**
    - Analiza el contenido de la respuesta HTTP en busca de una lista de indicadores comunes de errores SQL (ej., "syntax error", "mysql_fetch_array()", "ORA-"). La presencia de estos indicadores sugiere una posible vulnerabilidad.
    - Maneja excepciones de red y HTTP (errores de conexión, timeout, errores de estado HTTP 4xx/5xx) para una ejecución robusta.
- **Generación de Reporte:**
    - Imprime un informe detallado en la consola, mostrando cada URL probada y el resultado de la prueba.
    - Al final del proceso, presenta un resumen de las URLs identificadas como potencialmente vulnerables, facilitando la identificación de los objetivos que requieren una investigación más profunda.



4. Conclusión:

El script detector_sql_injection.py cumple eficazmente con los objetivos del ejercicio al automatizar la detección inicial de vulnerabilidades de SQL Injection. Proporciona una base sólida para entender cómo se realizan estas pruebas y cómo se identifican los posibles puntos débiles en las aplicaciones web, todo ello manteniendo un enfoque ético y seguro. Es una herramienta valiosa para la fase de reconocimiento básico y análisis en cualquier auditoría de seguridad.